

UNITED STATES PATENT AND TRADEMARK OFFICE

---



BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES

---

*EX PARTE* ANDREAS GUSTAFSSON

APPLICATION NO. 10/608,724

FILING DATE: JULY 26, 2003

TECHNOLOGY CENTER: 2186

---

BRIEF ON APPEAL

---

CARR & FERRELL LLP  
2200 GENG ROAD  
PALO ALTO, CA 94303  
T: 650.812.3400  
F: 650.812.3444

ATTORNEY FOR THE APPELLANT  
AND REAL-PARTY-IN-INTEREST

## TABLE OF CONTENTS

REAL-PARTY-IN-INTEREST .....	3
RELATED APPEALS AND INTERFERENCES.....	3
STATUS OF THE CLAIMS .....	3
STATUS OF AMENDMENTS .....	4
SUMMARY OF THE CLAIMED SUBJECT MATTER .....	6
Independent Claim 1 .....	6
Independent Claim 11 .....	7
Independent Claim 13 .....	8
Dependent Claim 14 .....	10
Dependent Claim 15 .....	11
Independent Claim 17 .....	12
Independent Claim 19 .....	13
Independent Claim 26 .....	15
Independent Claim 33 .....	16
Independent Claim 40 .....	17
GROUND OF REJECTION TO BE REVIEWED ON APPEAL .....	20
ARGUMENT.....	21
REJECTION OF CLAIMS 1-9 AND 11-43 ( <i>FLETCHER</i> IN VIEW OF <i>VISHIN</i> ) .....	21
Claims 1, 11, 13, 17, 26, 34 and 38-39 .....	21
Claims 2, 19, 25, 29, and 35 .....	36
Claims 3, 5, 20-23, and 27-28 .....	37
Claim 4 .....	41
Claims 6-9, 12, 16, 18, 24, 31-32, and 41-43 .....	42
Claims 14-15 and 30 .....	43
Claims 33, 36-37, and 40 .....	45
REJECTION OF CLAIM 10 ( <i>FLETCHER</i> IN VIEW OF <i>VISHIN</i> , FURTHER IN VIEW OF <i>RAMANATHAN</i> ) .....	46
CLAIMS APPENDIX .....	48
EVIDENCE APPENDIX .....	56
RELATED PROCEEDINGS APPENDIX .....	57

**REAL-PARTY-IN-INTEREST**  
**(37 C.F.R. § 41.37(C)(1)(i))**

The named Appellant in the present appeal is Andreas Gustafsson. The real-party-in-interest is Nominum, Inc. The Appellant has assigned the application to Nominum, Inc. The assignment is recorded in the U.S. Patent and Trademark Office at Reel No. 014231, Frame No. 0350.

**RELATED APPEALS AND INTERFERENCES**  
**(37 C.F.R. § 41.37(C)(1)(ii))**

The Appellant, the real-party-in-interest, and their agents and representatives are unaware of any related appeals and interferences that are concluded, ongoing, or otherwise prospective as of the date of submission of this BRIEF ON APPEAL.

**STATUS OF THE CLAIMS**  
**(37 C.F.R. § 41.37(C)(1)(iii))**

Claims 1-43 are presently pending. All of the presently pending claims have been (at least) twice rejected.

Claims 1-43 are being appealed.

Claims 1-9 and 11-43 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. patent application publication number 2002/0178238 to Fletcher et al. (hereinafter *Fletcher*) in view of U.S. patent number 5,860,146 to Vishin et al. (hereinafter *Vishin*).

Claim 10 is rejected under 35 U.S.C. § 103(a) as being unpatentable over *Fletcher* in view of *Vishin*, further in view of U.S. patent number 6,182,136 to Ramanathan et al. (hereinafter *Ramanathan*).

STATUS OF AMENDMENTS  
(37 C.F.R. § 41.37(C)(1)(iv))

U.S. patent application number 10/608,724 (Application) was filed on June 26, 2003.

A non-final office action mailed September 16, 2005 (September 2005 Office Action) indicated the pendency of claims 1-42.

An amendment filed December 15, 2005 (December 2005 Response) was responsive to objections to the drawings, objections to claims 10, 18, and 25, and the rejections presented by the Examiner in the September 2005 Office Action.

A second non-final office action mailed January 17, 2006 (January 2006 Office Action) indicated the pendency of claims 1-42. The Appellant believes that the remarks presented in the December 2005 Response were considered based on the new grounds of rejection presented in the January 2006 Office Action.

An amendment filed May 12, 2006 (May 2006 Response) was responsive to objections to claims 10 and 18, and the rejections presented by the Examiner in the January 2006 Office Action.

A final action mailed August 15, 2006 (August 2006 Final Action) indicated the pendency of claims 1-42.

An amendment filed October 4, 2006 (October 2006 Response) was responsive to the rejections maintained by the Examiner in the August 2006 Final Action.

A third non-final office action mailed November 16, 2006 (November 2006 Office Action) indicated the pendency of claims 1-42.

An amendment filed February 16, 2007 (February 2007 Response) added new claim 43 and was responsive to the rejections maintained by the Examiner in the November 2006 Office Action.

A Notice of Non-Compliant Amendment (March 2007 Notice of Non-Compliant Amendment) was mailed on March 27, 2007 indicating that the status identifier of new claim 43 was incorrect.

A Response to Notice of Non-Compliant Amendment filed April 3, 2007 (April 2007 Response) corrected the status identifier of claim 43.

A second final office action mailed June 8, 2007 (June 2007 Final Action) indicated the pendency of claims 1-43.

An amendment filed August 8, 2007 (August 2007 Response) was responsive to the rejections maintained by the Examiner in the June 2007 Final Action.

An advisory action mailed August 16, 2007 (August 2007 Advisory Action) indicated that the August 2007 Response had been entered. The Appellant believes that the remarks presented in the August 2007 Response were entered based on the claims indicated as having been entered and examined in the August 2007 Advisory Action.

Claims 1-43 are presented for appeal.

SUMMARY OF THE CLAIMED SUBJECT MATTER  
(37 C.F.R. § 41.37(C)(1)(v))

*Independent Claim 1*

Claim 1 as presented for appeal recites:

A caching server comprising:  
    an answer cache configured to access answer information  
        through a flat data structure;  
    a referral cache configured to store referral information;  
    and  
    computer instructions configured to translate a domain  
        name into DNS information by examining the  
        answer cache and, responsive to the results of  
        examining the answer cache, examining the referral  
        cache.

The claimed caching server generally concerns the caching of domain names and Domain Name System (DNS) information. This caching server generally corresponds with the description set forth in the SUMMARY OF THE INVENTION as it pertains to “a caching server having a segregated cache.” Application, [0009]; *see also* [0010].<sup>1</sup> The caching server is shown in FIG. 1 and is identified by reference number 160.

Claim 1 additionally recites an “an answer cache configured to access answer information through a flat data structure.” The “answer cache” of claim 1 is disclosed in the patent application at [0010] and [0022]. For example, [0010] recites, “a caching server comprising an answer cache configured to access information through a flat data structure. Additionally, [0022] recites, “The answer information stored in the answer cache is stored in a flat data structure.” The answer cache is shown in FIG. 1 and is identified by reference number 165.

Claim 1 also recites “a referral cache configured to store referral information.” A referral cache configured to store referral information is supported by the disclosure at [0010]—a “referral cache configured to store referral information”—as well as at

---

<sup>1</sup> All references to the Application are exemplary and are not intended to be limiting. Other support may exist. The present references are made solely to satisfy the requirements of 37 C.F.R. 41.37(c)(1)(v).

[0022]—a “referral cache . . . configured to store referral information including referrals to authoritative servers.” The referral cache is shown in FIG. 1 and is identified by reference number 168.

Claim 1 further recites “computer instructions configured to translate a domain name into DNS information by examining the answer cache and, responsive to the results of examining the answer cache, examining the referral cache.” The computer instructions are supported by the disclosure at [0010] which states, “computer instructions [are] configured to determine DNS information by examining the answer cache and, responsive to the answer cache, examining the referral cache.” The computer instructions are shown in FIG. 1 and are identified by reference number 163.

***Independent Claim 11***

Claim 11 as presented for appeal recites:

A computer readable medium having stored thereupon computer code configured to determine DNS information associated with a domain name, the computer code comprising:

- a code segment configured to receive a request for the DNS information corresponding to a domain name;
- a code segment configured to examine a first cache to find the DNS information, the first cache including a flat data structure and configured to store the DNS information or a pointer to the DNS information; and
- a code segment configured to initiate a search of a second cache if the DNS information is not found by examining the first cache, the second cache configured to store data referring to further locations on a computer network wherein the DNS information may be found.

The claimed computer readable medium generally concerns determining DNS information associated with a domain name. The computer readable medium is, at least, disclosed in the Application at [0011] which discloses:

Various embodiments of the invention include a computer readable medium having stored thereupon computer code configured for determining DNS information associated with a domain name, the computer code comprising a code segment configured for receiving a request for the DNS information corresponding to a domain name, a code segment configured for examining a first cache to find the DNS

information, the first cache including a flat data structure and configured to store the DNS information or a pointer to the DNS information, and a code segment configured to initiate a search of a second cache if the DNS information is not found by examining the first cache, the second cache configured to store data referring to further locations on a computer network wherein the DNS information may be found.

The “code segment configured to receive a request for the DNS information corresponding to a domain name” is also supported in the Application by FIG. 2, step 210, “receive data request.”

The “code segment configured to examine a first cache to find the DNS information, the first cache including a flat data structure and configured to store the DNS information or a pointer to the DNS information” is also supported in the Application by FIG. 2, step 220, “Examine Answer Cache.”

The “code segment configured to initiate a search of a second cache if the DNS information is not found by examining the first cache, the second cache configured to store data referring to further locations on a computer network wherein the DNS information may be found” is also supported in the Application by FIG. 2, steps 230 and 250.

### ***Independent Claim 13***

Claim 13 as presented for appeal recites:

A computer network comprising:

- means for receiving a request for DNS information corresponding to a domain name;
- means for examining a first cache to find the DNS information, the first cache configured to store the DNS information or a pointer to the DNS information; and
- means for searching a second cache if the DNS information is not found by examining the first cache, the second cache configured to store data referring to further locations on the computer network wherein the DNS information may be found.



The claimed computer network generally concerns providing DNS services using separate answer and referral caches. Application, title. *See also* [0012] FIG. 1, Caching Server 160, and FIG. 2.

The “means for receiving a request for DNS information corresponding to a domain name” of claim 13 is in means plus function form as permitted by 35 U.S.C. 112, sixth paragraph. The structure described in the specification corresponding to the claimed function includes the caching server 160 of FIG. 1. For example, the “[c]aching server 160 includes Computer Instruction 163 configured to receive the query.”

Application, [0025]. Further, means are described with respect to FIG. 2, step 210. From the specification:

Caching Server 160 receives a request for an IP address, or other DNS information, corresponding to a specific domain name, in a Receive Data Request Step 210. This request is typically received from Client 110 or from some other component of Computer Network 100. The request includes the specific domain name, for which the associated DNS information is desired, and a type of the DNS information expected. This type is optionally an IP address, MX record, or the like.

[0029].

The “means for examining a first cache to find the DNS information, the first cache configured to store the DNS information or a pointer to the DNS information” of claim 13 is in means plus function form as permitted by 35 U.S.C. 112, sixth paragraph. The structure described in the specification corresponding to the claimed function includes the Computer Instructions 163 of FIG. 1. For example, “Computer Instructions 163 are configured to first look for answer information in an Answer Cache 165 for DNS information associated with the domain name.” Application, [0026]. Further, means are described with respect to FIG. 2, step 220. From the specification:

In an Examine Answer Cache Step 220, Computer Instructions 163 are used to look in Answer Cache 165 for the desired DNS information. In embodiments wherein Answer Cache 165 includes a hash table, the time required for the examination is approximately independent of the size of Answer Cache 165 and of a number of labels in the specified domain name.

[0030].

The “means for searching a second cache if the DNS information is not found by examining the first cache, the second cache configured to store data referring to further locations on the computer network wherein the DNS information may be found” of claim 13 is in means plus function form as permitted by 35 U.S.C. 112, sixth paragraph. The structure described in the specification corresponding to the claimed function includes the Computer Instructions 163 of FIG. 1. For example, “Computer Instructions 163 are configured to next look in Referral Cache 168 for referral information referring to other components of Computer Network 100 that may have information leading to the desired DNS information.” [0027]. Further, means are described with respect to FIG. 2, steps 230 and 250. From the specification:

In a Found Step 230, Computer Instructions 163 determine if the desired DNS information was found in Examine Answer Cache Step 220. . . . If not, then the method proceeds to a Search Referral Cache Step 250.

In Search Referral Cache Step 250, Computer Instructions 163 initiate a search of Referral Cache 168 for referral information referring to one or more Authoritative Server 170 and associated with the domain name to be translated to DNS information. This search is configured to identify referral information associated with the parent domain name, available in the cache, that is closest to the domain name to be translated. [0031], [0032].

#### ***Dependent Claim 14***

Claim 14 as presented for appeal recites, “The computer network of claim 13, further including means for storing data in the first cache such that a time required to examine the first cache is essentially constant as a function of a number of labels comprising the domain name.”

The “means for storing data in the first cache such that a time required to examine the first cache is essentially constant as a function of a number of labels comprising the domain name” of claim 14 is in means plus function form as permitted by 35 U.S.C. 112, sixth paragraph. The structure described in the specification corresponding to the claimed function includes the Answer Cache 165 of FIG. 1. For example:

Answer Cache 165 is configured to include a flat table, such as a hash table. In embodiments wherein Answer Cache 165 is a hash table, a time required to look for the answer information remains essentially constant as a function of the number of labels in the domain name being translated.

[0026]. Further, means are described with respect to FIG. 2, step 220. From the specification, “[i]n embodiments wherein Answer Cache 165 includes a hash table, the time required for the examination is approximately independent . . . of a number of labels in the specified domain name.” [0030].

#### *Dependent Claim 15*

Claim 15 as presented for appeal recites, “The computer network of claim 13, further including means for storing data in the first cache such that a time required to examine the first cache is essentially constant as a function of a size of the first cache.”

The “means for storing data in the first cache such that a time required to examine the first cache is essentially constant as a function of a size of the first cache” of claim 15 is in means plus function form as permitted by 35 U.S.C. 112, sixth paragraph. The structure described in the specification corresponding to the claimed function includes the Answer Cache 165 of FIG. 1. For example:

Answer Cache 165 is configured to include a flat table, such as a hash table. In embodiments wherein Answer Cache 165 is a hash table, a time required to look for the answer information remains essentially constant as a function of the number of labels in the domain name being translated. In some embodiments the time required to look for the answer information is approximately independent of the size of Answer Cache 165.

[0026]. Further, means are described with respect to FIG. 2, step 220. From the specification, “[i]n embodiments wherein Answer Cache 165 includes a hash table, the time required for the examination is approximately independent of the size of Answer Cache 165.” [0030].

### *Independent Claim 17*

Claim 17 as presented for appeal recites:

A computer network comprising:

- a computing system configured to access a component of the computer network using a domain name;
- a caching server including a first data structure configured for translating the domain name into DNS information, and means for examining the first data structure in a time that is essentially constant as a function of a number of labels comprising the domain name; and
- a second data structure configured for translating the domain name into DNS information.

The claimed computer network generally concerns translating domain names into DNS information. Application, abstract. *See also* [0013], FIG. 1, and FIG. 2.

Claim 17 recites, “a computing system configured to access a component of the computer network using a domain name.” The computing system of claim 17 is supported, at least, by FIG. 1 which depicts a Client 110 including Access Software 130. For example, “Client 110 includes Access Software 130 configured to identify and access a network location, such as Web Server 140, using a domain name.” Application, [0025].

Claim 17 additionally recites, “a caching server including a first data structure configured for translating the domain name into DNS information.” The caching server is supported, at least, by FIG. 1 which depicts the Caching Server 160. For example, the specification includes “translating a domain name into DNS information such as IP addresses, MX records, or other DNS data types. In various embodiments, . . . one or more caching servers each having a segregated cache divided into an answer cache and a referral cache.” [0022].

Further, the caching server of claim 17 includes “means for examining the first data structure in a time that is essentially constant as a function of a number of labels comprising the domain name” which is in means plus function form as permitted by 35 U.S.C. 112, sixth paragraph. The structure described in the specification corresponding to the claimed function includes the Computer Instructions 163 of FIG. 1. For example, “Computer Instructions 163 are configured to first look for answer information in an

Answer Cache 165 for DNS information associated with the domain name.”

Application, [0026]. Further, means are described with respect to FIG. 2, step 220. From the specification:

In an Examine Answer Cache Step 220, Computer Instructions 163 are used to look in Answer Cache 165 for the desired DNS information. In embodiments wherein Answer Cache 165 includes a hash table, the time required for the examination is approximately independent of the size of Answer Cache 165 and of a number of labels in the specified domain name.

[0030].

Claim 17 also includes “a second data structure configured for translating the domain name into DNS information.” The second data structure is supported, at least, by FIG. 1 which depicts the Caching Server 160 including an Answer Cache 163 and a Referral Cache 165. For example, the specification includes “translating a domain name into DNS information such as IP addresses, MX records, or other DNS data types. In various embodiments, . . . one or more caching servers each having a segregated cache divided into an answer cache and a referral cache.” [0022].

### ***Independent Claim 19***

Claim 19 as presented for appeal recites:

A method of determining DNS information, the method comprising:

- receiving a request for DNS information corresponding to a domain name;
- examining an answer cache for answer information, the answer cache including a hash table configured to store the answer information or to store a pointer to the answer information; and
- searching a tree data structure if the DNS information is not found by examining the answer cache.

The claimed method generally concerns determining DNS information. Application, title. *See also* [0014], FIG. 1, and FIG. 2.

Claim 19 recites, “receiving a request for DNS information corresponding to a domain name.” For example, the “[c]aching server 160 includes Computer Instruction

163 configured to receive the query.” Application, [0025]. Further, as described with respect to FIG. 2, step 210:

Caching Server 160 receives a request for an IP address, or other DNS information, corresponding to a specific domain name, in a Receive Data Request Step 210. This request is typically received from Client 110 or from some other component of Computer Network 100. The request includes the specific domain name, for which the associated DNS information is desired, and a type of the DNS information expected. This type is optionally an IP address, MX record, or the like.  
[0029].

Claim 19 further recites, “examining an answer cache for answer information, the answer cache including a hash table configured to store the answer information or to store a pointer to the answer information.” For example, “Computer Instructions 163 are configured to first look for answer information in an Answer Cache 165 for DNS information associated with the domain name.” Application, [0026]. Further, a method step is shown in FIG. 2, step 220. From the specification:

In an Examine Answer Cache Step 220, Computer Instructions 163 are used to look in Answer Cache 165 for the desired DNS information. In embodiments wherein Answer Cache 165 includes a hash table, the time required for the examination is approximately independent of the size of Answer Cache 165 and of a number of labels in the specified domain name.

[0030]. Further, the specification includes, “examining a first cache to find the DNS information, the first cache including a flat data structure and configured to store the DNS information or a pointer to the DNS information.” Application, [0011]

Claim 19 also recites “searching a tree data structure if the DNS information is not found by examining the answer cache.” See, e.g., FIG. 2, steps 230 and 250. For example, “initiat[ing] a search of a second cache if the DNS information is not found by examining the first cache, the second cache configured to store data referring to further locations on a computer network wherein the DNS information may be found.”

Application at [0011]. Further, “referral information stored in the referral cache are stored in a tree-based or other hierarchical structure.” Application [0022].

***Independent Claim 26***

Claim 26 as presented for appeal recites:

A method of determining DNS information, the method comprising:  
receiving a request for DNS information corresponding to a domain name;  
examining an answer cache to find answer information, responsive to the received request, the answer cache including a flat data structure; and  
responsive to the examination of the answer cache, searching a referral cache.

The claimed method generally concerns determining DNS information. Application, title. *See also* [0015], FIG. 1, and FIG. 2.

Claim 26 recites, "receiving a request for DNS information corresponding to a domain name." For example, the "[c]aching server 160 includes Computer Instruction 163 configured to receive the query." Application, [0025]. Further, as described with respect to FIG. 2, step 210:

Caching Server 160 receives a request for an IP address, or other DNS information, corresponding to a specific domain name, in a Receive Data Request Step 210. This request is typically received from Client 110 or from some other component of Computer Network 100. The request includes the specific domain name, for which the associated DNS information is desired, and a type of the DNS information expected. This type is optionally an IP address, MX record, or the like.  
[0029].

Claim 26 further recites, "examining an answer cache to find answer information, responsive to the received request, the answer cache including a flat data structure." For example, "Computer Instructions 163 are configured to first look for answer information in an Answer Cache 165 for DNS information associated with the domain name."

Application, [0026]. From the specification:

In an Examine Answer Cache Step 220, Computer Instructions 163 are used to look in Answer Cache 165 for the desired DNS information. In embodiments wherein Answer Cache 165 includes a hash table, the time required for the examination is approximately independent of the size of Answer Cache 165 and of a number of labels in the specified domain name.

[0030]; *see also* FIG. 2, step 220. Further, the specification includes, at least, "examining a first cache to find the DNS information, the first cache including a flat data structure and

configured to store the DNS information or a pointer to the DNS information.”

Application, [0011].

Claim 26 further recites “responsive to the examination of the answer cache, searching a referral cache.” See, e.g., FIG. 2, steps 230 and 250. For example, “initiat[ing] a search of a second cache if the DNS information is not found by examining the first cache, the second cache configured to store data referring to further locations on a computer network wherein the DNS information may be found.” Application, [0011].

### ***Independent Claim 33***

Claim 33 as presented for appeal recites:

A method of storing data in a cache, the method comprising:

- requesting DNS information;
- receiving data in response to the request;
- classifying the response received; and
- storing the data received in either a referral cache or an answer cache based on the classification.

The claimed method for storing data in a cache may be useful in providing DNS services using separate answer and referral caches. Application, title. See also [0016] and FIG. 3.

Claim 33 recites, “requesting DNS information.” For example, “specifically, in a Query Authoritative Server Step 260, Caching Server 160 requests DNS information associated with a specific domain name. This request includes the specific domain name and is made to an authoritative server, such as Authoritative Server 170.” Application, [0035].

Claim 33 also includes, “receiving data in response to the request.” For example, “In a Receive Response Step 320, included in various embodiments of Process Response Step 270, Caching Server 160 receives a response to the request made in, for example, Query Authoritative Server Step 260. This response typically includes data such as answer information, referral information, error information, or the like.” Application, [0036].



Additionally, claim 33 states, “classifying the response received” and “storing the data received in either a referral cache or an answer cache based on the determination.” For example,

In a Determine Response Type Step 330, Computer Instructions 163 determine the type of response received in Receive Response Step 320. If the response is a referral response, then the method proceeds to a Store in Referral Cache Step 340. If the type is an answer response, then the method proceeds to a Store in Answer Cache Step 350.

Application, [0039].

#### ***Independent Claim 40***

Claim 40 as presented for appeal recites:

A method of caching DNS information, the method comprising:  
requesting DNS information;  
receiving data in response to requesting DNS information;  
classifying the response received as an answer response or a referral response;  
storing the response received in either a referral cache or an answer cache based on the classification, the answer cache including a flat data structure;  
receiving a request for DNS information corresponding to a domain name;  
examining the answer cache to find answer information, responsive to the received request; and  
responsive to the examination of the answer cache, searching the referral cache.

The claimed method of caching DNS information generally concerns providing DNS services using separate answer and referral caches. Application, title. *See also* [0017] and FIG. 3.

Claim 40 recites “requesting DNS information.” For example, “specifically, in a Query Authoritative Server Step 260, Caching Server 160 requests DNS information associated with a specific domain name. This request includes the specific domain name and is made to an authoritative server, such as Authoritative Server 170.” Application, [0035].

Claim 40 also includes, “receiving data in response to the request.” For example, “In a Receive Response Step 320, included in various embodiments of Process Response Step 270, Caching Server 160 receives a response to the request made in, for example, Query Authoritative Server Step 260. This response typically includes data such as answer information, referral information, error information, or the like.” Application, [0036].

Additionally, claim 40 states, “classifying the response received as an answer response or a referral response” and “storing the data received in either a referral cache or an answer cache based on the determination, the answer cache including a flat data structure.” For example,

In a Determine Response Type Step 330, Computer Instructions 163 determine the type of response received in Receive Response Step 320. If the response is a referral response, then the method proceeds to a Store in Referral Cache Step 340. If the type is an answer response, then the method proceeds to a Store in Answer Cache Step 350.

Application, [0039]. Also, “answer information stored in the answer cache is stored in a flat data structure, such as a hash table.” Application, [0022].

Claim 40 recites, “receiving a request for DNS information corresponding to a domain name.” For example, the “[c]aching server 160 includes Computer Instruction 163 configured to receive the query.” Application, [0025]. Further, as described with respect to FIG. 2, step 210:

Caching Server 160 receives a request for an IP address, or other DNS information, corresponding to a specific domain name, in a Receive Data Request Step 210. This request is typically received from Client 110 or from some other component of Computer Network 100. The request includes the specific domain name, for which the associated DNS information is desired, and a type of the DNS information expected. This type is optionally an IP address, MX record, or the like.  
[0029].

Claim 40 further recites, “examining an answer cache to find answer information, responsive to the received request, the answer cache including a flat data structure.” For example, “Computer Instructions 163 are configured to first look for answer information

in an Answer Cache 165 for DNS information associated with the domain name.”

Application, [0026]. Further, a method step is shown in FIG. 2, step 220. From the specification:

In an Examine Answer Cache Step 220, Computer Instructions 163 are used to look in Answer Cache 165 for the desired DNS information. In embodiments wherein Answer Cache 165 includes a hash table, the time required for the examination is approximately independent of the size of Answer Cache 165 and of a number of labels in the specified domain name.

[0030].

Claim 40 further recites “responsive to the examination of the answer cache, searching a referral cache.” See, *e.g.*, FIG. 2, steps 230 and 250. For example, “initiat[ing] a search of a second cache if the DNS information is not found by examining the first cache, the second cache configured to store data referring to further locations on a computer network wherein the DNS information may be found.” Application, [0011].

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

(37 C.F.R. § 41.37(C)(1)(vi))

- I. Are claims 1-9 and 11-43 obvious under 35 U.S.C. § 103(a) over *Fletcher* in view of *Vishin*?
- II. Is claim 10 obvious under 35 U.S.C. § 103(a) over *Fletcher* in view of *Vishin*, further in view of *Ramanathan*?

ARGUMENT  
(37 C.F.R. § 41.37(C)(1)(vii))

I. REJECTION OF CLAIMS 1-9 AND 11-43 (*FLETCHER* IN VIEW OF *VISHIN*)

A. Claims 1, 11, 13, 17, 26, 34 and 38-39

1. Summary of Prosecution with Respect to Claims 1, 11, 13, 17, 26, 34 and 38-39

In the November 2006 Office Action, the Examiner rejected claims 1, 11, 13, 17, 26, 34 and 38-39. The Examiner based the rejection on an analysis of independent claim 1. Page 2. Claims 11, 13, 17, 26, 34 and 38-39 were “also rejected based on the same rationale as claim 1.” November 2006 Office Action, 3. Claim 1 recites:

1. A caching server comprising:
  - an answer cache configured to access answer information through a flat data structure;
  - a referral cache configured to store referral information;
  - and
  - computer instructions configured to translate a domain name into DNS information by examining the answer cache and, responsive to the results of examining the answer cache, examining the referral cache.

The Examiner contended that *Fletcher* disclosed an answer cache, a referral cache, and computer instructions. See November 2006 Office Action, 2-3 (citing [0008] of *Fletcher*). The Examiner admitted that “*Fletcher* does not clarify that the answer cache stored answer information in a flat data structure.” November 2006 Office Action, 3. The Examiner asserted that “*Vishin* . . . teaches a computer system which includes a translation lookaside buffer (TLB) (i.e. 122 in FIG. 5), similar to claimed answer cache, for storing the address information for the local page table entries.” November 2006 Office Action, 3. The Examiner also stated, without support, “By using the hash table (i.e. the flat data structure) in [the] answer cache as taught by *Vishin*, it reduces the number of memory accesses and as a result of that, it is faster than the lookup in the tree structure.” November 2006 Office Action, 3.

In response, the Applicant (now Appellant) argued in the April 2007 Response that (1) *Vishin* does not teach an answer cache having a flat data structure:

The RTL16 of *Vishin* is a table of partial physical memory addresses used to translate a physical memory address into a remote physical memory address, (*Vishin* Col. 4 lines 42-62). In contrast, the "answer cache" recited in Claim 1 is "an answer cache configured to access answer information through a flat data structure." One of ordinary skill in the art would understand "answer information" to include, for example, an Internet Protocol (IP) address provided in response to DNS request. One of ordinary skill would not expect "answer information" as received from an "answer cache" to include "a remote physical memory address." An IP address and a physical memory address are fundamentally different non-interchangeable things. Specifically, a physical memory address is a hardware address of a memory location within a memory device, while an IP address is an address used by network elements to communicate data packets according to IP standards. A physical memory address could not be used in place of an IP address, and does not appear to be provided as an answer to a DNS request. As such, the RTL16 of *Vishin* is not equivalent to the "answer cache" recited in Claim 1.

page 11; (2) the combination suggested by the Examiner does not have a reasonable expectation of success:

As recited in lines 5-7 of Claim 1, the "answer cache" of Claim 1 can be examined by computer instructions to "translate a domain name into DNS information." It is unclear to the Applicant how the RTL16 of *Vishin* could be used to translate a domain name into DNS information. As such, the proposed combination does not appear to have a reasonable expectation of success. The Applicant, therefore, requests that the Examiner explain how a table of partial physical memory addresses intended for use in translating a physical memory address into a remote physical memory address, could be used to "translate a domain name into DNS information," as recited in Claim 1, or allow Claim 1 and those claims that depend therefrom.

page 12; (3) the combination suggested by the Examiner requires substantial modifications to the elements of claim 1 without support suggestion from the specification of either:

in combining the teachings of *Fletcher* and *Vishin*, the Examiner appears to be doing more than merely combining elements, e.g., the Examiner is doing more than modifying *Fletcher* by addition of an element taught in *Vishin*. Specifically, the Examiner appears to be suggesting that the teachings of *Vishin* be changed substantially in manners unsupported by

either reference in order to fit the needs of the Examiner's rejection, and that these changed teachings be added to *Fletcher*. While the combination of elements may be reasonable under §103(a), making substantial modifications to these elements, without support or suggestion from the specification of either, in order for these elements to function together is not proper.

page 12; (4) the Examiner has not provided a sufficient motivation to combine under 35 U.S.C. §103(a), and more specifically, that *Vishin* does not teach a "hash table" as asserted by the Examiner:

As a motivation to combine the cited art, the Examiner states, "[b]y using the hash table (i.e. the flat data structure) in answer cache as taught by *Vishin*, it reduces the number of memory accesses and as a result of that, it is faster than the lookup in the tree structure."

The Applicant traverses the Examiner's statement on the grounds that *Vishin* does not appear to teach a "hash table." The Applicant is unable to find any teaching within *Vishin* that the RTL60 is a "hash table." Since *Vishin* does not teach the use of a "hash table," any speed advantage that could theoretically be achieved therefrom cannot provide the basis for combining *Fletcher* and *Vishin*. . . .

Further, the Applicant is unable to find any support within the cited art that use of the RTL60 table of *Vishin* "reduces the number of memory accesses" as suggested by the Examiner. To the contrary, the use of a table for translating a physical memory address would appear to result in a greater number of memory accesses because the RTL60 table must be accessed first, rather than accessing the physical memory directly.

page 13; (5) the Examiner does not provide any evidence that the suggested motivation would be known to one of ordinary skill in the art at the time of the invention:

the Examiner appears to have defined a problem (lacking a hash table) in terms of its solution (adding a hash table). As stated in *In re Beattie*, 974 F.2d 1309, 1312 (Fed. Cir. 1992) "[d]efining the problem in terms of its solution reveals improper hindsight in the selection of the prior art relevant to obviousness." The Applicant, therefore, requests that the Examiner provide either a motivation to combine the cited art from within the cited art or other evidence that the suggested motivation would be known to one of ordinary skill in the art at the time of the invention.

page 14; and (6) the cited references are in substantially different fields of art:

The claimed invention is in the field "network communications," (Application as filed, paragraph [0001]), and more specifically some

claimed embodiments are in the fields of Internet Protocol and Domain Name Systems. In contrast, *Vishin* relates to “virtual memory management subsystems, and particularly to a memory controller that manages access to remote physical addresses,” (Col. 1 lines 6-8). The Applicant maintains that the fields of network communications and virtual memory management are substantially different. The first deals with long range software managed communications, while the second deals with physically managed local memory access. It is, therefore, the position of the applicant that the one of ordinary skill in the art of network communications would not look to the teachings of *Vishin* to modify the teachings of *Fletcher*.  
page 15.

The Examiner’s June 2007 Final Action substantively reiterated the previously rendered rejection; this action was deemed final. In the aforementioned June 2007 Final Action, the Examiner stated that “Applicant’s arguments filed on April 5, 2007 have been fully considered but they are not deemed persuasive.” Page 2. The Examiner dismissed the six arguments raised by the Appellant in the remarks. June 2007 Final Action, 9-10. In response to the first argument, the Examiner stated, “*Vishin*, however, does teach that the RTL160 is a buffer which stores (address) information in [a] flat data structure.” Page 9. The Examiner, however, did not provide a reference to any teaching in *Vishin* to support this argument. In response to the second argument, the Examiner stated, “*Fletcher* does teach that computer instructions configured to translate a domain name into DNS information by examining the answer cache (e.g., see paragraphs [0005] and [0008]). June 2007 Final Action, 9. In response to the third, fourth, and fifth arguments, the Examiner merely stated, “the motivation for the rejection is found in the knowledge generally available to one of ordinary skill in the art” without further support or clarification. June 2007 Final Action, 9. In response to the sixth argument, the Examiner asserted that *Vishin* “discloses about translating address information and therefore, can be used in the *Fletcher* prior art to translate and store the address information.” June 2007 Final Action, 9-10.

In response, the Appellant, in the August 2007 Response, repeated the remarks in the April 2007 Response and responded, (1) with regard to the Examiner’s assertion that the RTL160 is a buffer which stores address information in a flat data structure, that:



The Applicant has reviewed the portions of *Vishin* previously cited by the Examiner, notably the abstract and FIG. 5, as well as the rest of the patent and is still unable to identify a hash table or other flat data structure in the cited reference. The Applicant, therefore, repeats the request that the Examiner specifically point out a teaching that the RTL 160 of *Vishin* is a "hash table."

page 12; (2) with regard to the Examiner's assertion that *Fletcher* teaches computer instructions configured to translate a domain name into DNS information by examining the answer cache, that:

because the Examiner has merely relied on the purported teaching in *Vishin* of a hash table as evidence of teaching "an answer cache configured to access answer information through a **flat data structure**" (emphasis added), the Examiner has not meaningfully addressed the Applicant's argument that it is "unclear to the Applicant how the RTL 160 of *Vishin* could be used to translate a domain name into DNS information."

page 13; (3)-(5) with regard to the Examiner's assertion that the motivation for the rejection is found in the knowledge generally available to one of ordinary skill in the art, that:

The Applicant respectfully refers the Examiner to the Memorandum of May 3, 2007 by Margaret A. Focarino to the Technology Center Directors regarding the "Supreme Court decision on *KSR Int'l Co., v. Teleflex, Inc.*" In the memorandum, Ms Focarino concludes, "Therefore, in formulating a rejection under 35 U.S.C. §103(a) based upon a combination of prior art elements, it remains necessary to identify **the reason why** a person of ordinary skill in the art would have combined the prior art elements in the manner claimed." Page 2 (emphasis added). As has been held by the Courts, "to establish a prima facie case, the USPTO may not rely on unsupported assertions about the level of ordinary skill in the art or bare conclusions that one of ordinary skill could apply such skill to obtain the claimed invention." In *re Sun*, 31 USPQ2d 1451, 1456 (Fed. Cir. 1994)(Mayer, J. concurring).

The Applicant respectfully avers that a motivation "found in the knowledge generally available to one of ordinary skill in the art," does not meet the threshold requirement to provide a prima facie case for a rejection under 35 U.S.C. §103(a) according to current practice. More specifically, the Examiner has not provided a **reason why** the knowledge generally available to one of ordinary skill in the art would motivate such a person to make the proposed combination.

pages 15-16; (6) with regard to the Examiner's assertion that *Vishin* discloses translating address information and can be used in *Fletcher* to translate and store the address information, that:

the addresses disclosed in *Fletcher* and the present application are substantially different from those disclosed in *Vishin*. The addresses described in the present application and in *Fletcher* are IP addresses associated with a device on a communications network. See, e.g., *Fletcher* [0005] and Specification [0002]. In contrast, the addresses described by *Vishin* are stored in "a primary translation lookaside buffer for . . . translating virtual addresses into physical address, local memory coupled to the data processor . . . and remotely located memory coupled to the data processor by a computer network." Abstract.

page 17. In light of at least the above, the Appellant respectfully requested that the Examiner withdraw the finality of the June 2007 Final Action.

In the August 2007 Advisory Action, the Examiner simply noted that "all the arguments were responded/answered in the last office action mailed out on June 8, 2007."

Regarding claims 11, 13, 17, 26, 34 and 38-39, the Examiner rejected these claims based on the same rationale as claim 1. November 2006 Office Action, 3.

The Appellant replied in the April 2006 Response that "Claims 11, 13, 17, 26, 34 and 38-39 include numerous limitations not included in Claim 1 and not addressed by the Examiner as pointed out previously." Pages 15-16. The Appellant elaborated:

For example, [in **claim 11**] the cited art does not appear to teach "*the first cache including a flat data structure and configured to store the DNS information or a pointer to the DNS information.*" Further, the cited art does not appear to teach "*a code segment configured to initiate a search of a second cache if the DNS information is not found by examining the first cache, the second cache configured to store data referring to further locations on a computer network wherein the DNS information may be found.*" . . .

**Claim 13** recites, in part, "*means for examining a first cache to find the DNS information,*" and "*means for searching a second cache . . . the second cache configured to store data referring to further locations on the computer network wherein the DNS information may be found.*" These limitations are not included in Claim 1 and do not appear to be taught by the cited art. For example, the cited art does not teach both a first cache that can be examined to find "*DNS information*" in combination with "*a second cache*

configured to store data referring to further locations on the computer network wherein the DNS information may be found." . . .

**Claim 17** recites, in part, "a caching server including a first data structure configured for translating the domain name into DNS information, and means for examining the first data structure in a time that is essentially constant as a function of a number of labels comprising the domain name." These limitations are not included in Claim 1 and do not appear to be taught by the cited art. For example, the cited art does not appear to teach "means for examining the first data structure in a time that is essentially constant as a function of a number of labels comprising the domain name." . . .

**Claims 26 and 34** are believed to be allowable for at least the same reasons as Claim 1.

**Claims 38 and 39** recite "wherein the answer cache is configured to store answer information and the referral cache is configured to store referral information." These limitations are not included in Claim 1 and do not appear to be taught by the cited art. Specifically, Claim 1 does not teach "the answer cache is configured to store answer information and the referral cache is configured to store referral information." As pointed out elsewhere herein, those features of *Vishin* that the Examiner suggests teaches "the answer cache" is taught to include "physical addresses" and not "answer information," as recited in Claims 38 and 39.

April 2006 Response, 16-18.

The Examiner did not address these limitations in the June 2007 Final Action or in the August 2007 Advisory Action. The Appellant repeated the above remarks in the August 2007 Response, 17-21. The present appeal followed; a notice of appeal having been filed September 10, 2007:

## 2. Present Arguments with Respect to Claims 1, 11, 13, 17, 26, 34 and 38-

39

### *a. Vishin Fails to Disclose an Answer Cache as asserted by the Examiner*

To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981 (CCPA 1974). The Appellant traversed the rejections of claim 1 at least because *Vishin* does not teach or suggest “an answer cache configured to access answer information through a flat data structure” as recited in claim 1. As such, the Examiner has failed to establish a *prima facie* case of obviousness.

The RTL60 of *Vishin* is a table, within a virtual memory system, of partial physical memory addresses used to translate a physical memory address into a remote physical memory address. Col. 4, l. 42-62. In contrast, the “answer cache” recited in Claim 1 is “an answer cache configured to access answer information through a flat data structure.” One of ordinary skill in the art would understand “answer information” to include, for example, an Internet Protocol (IP) address provided in response to DNS request. One of ordinary skill would not expect “answer information” as received from an “answer cache” to include “a remote physical memory address.” An IP address and a physical memory address are fundamentally different non-interchangeable things. Specifically, a physical memory address is a hardware address of a memory location within a memory device, while an IP address is an address used by network elements to communicate data packets according to IP standards. A physical memory address could not be used in place of an IP address, and does not appear to be provided as an answer to a DNS request. As such, the RTL60 of *Vishin* is not equivalent to the “answer cache” recited in Claim 1.

The Examiner responded that “*Vishin*, however, does teach that the RTL60 [remote translation lookaside buffer] is a buffer which stores (address) information in a flat data structure.” June 2007 Final Action, page 9. As such, the Examiner did not address the above argument. The Examiner is improperly equating the RTL60 of *Vishin*,

which accesses physical memory addresses, with the answer cache of claim 1 which accesses answer information in response to a DNS request.

Further, *Vishin* does not appear to teach a “hash table” or other flat data structure. The Appellant has reviewed the portions of *Vishin* previously cited by the Examiner, notably the abstract and FIG. 5, as well as the rest of the patent and is still unable to identify a hash table or other flat data structure in the cited reference. The Examiner did not respond to Appellant’s multiple requests for a specific teaching in *Vishin* of a “hash table” or “flat data structure.”

*b. Fletcher and Vishin are not analogous art*

The rejection under §103(a) is improper because the cited references are in substantially different fields of art, and as such one of ordinary skill in the art of the invention would not look to combine the features of *Vishin* with those of *Fletcher*.

In *In re Oetiker*, it was held that “the combination of elements from non-analogous sources, in a manner that reconstructs the applicant’s invention only with the benefit of hindsight, is insufficient to present a prima facie case of obviousness.” 977 F.2d at 1447. Further, MPEP §2141.01(a) provides that “[t]he examiner must determine what is ‘analogous prior art’ for the purpose of analyzing the obviousness of the subject matter at issue. In order to rely on a reference as a basis for rejection of an applicant’s invention, the reference must either be in the field of applicant’s endeavor or, if not, then be reasonably pertinent to the particular problem with which the inventor was concerned.” *In re Oetiker*, 977 F.2d 1443, 1446 (Fed. Cir. 1992).

The claimed invention is in the field “network communications,” (Application, [0001]), and more specifically some claimed embodiments are in the fields of Internet Protocol and Domain Name Systems. *Fletcher* is also within the field of network communications. In contrast, *Vishin* relates to “virtual memory management subsystems, and particularly to a memory controller that manages access to remote physical addresses,” (Col. 1 lines 6-8). The fields of network communications and virtual memory management are substantially different. The first deals with long range

software managed communications, while the second deals with physically managed local memory access. Therefore, one of ordinary skill in the art of network communications would not look to the teachings of *Vishin* to modify the teachings of *Fletcher*.

In response, the Examiner asserted that “the *Vishin* prior art discloses about translating address information and, therefore, can be used in the *Fletcher* prior art to translate and store the address information in the flat data structure as suggested by *Vishin*.” Page 10.

However, the addresses disclosed in *Fletcher* and the present application are substantially different from those disclosed in *Vishin*. The addresses described in the present application and in *Fletcher* are IP addresses associated with a device on a communications network. See, e.g., *Fletcher* [0005] and *Application*, [0002]. In contrast, the addresses described by *Vishin* are stored in “a primary translation lookaside buffer for . . . translating virtual addresses into physical address, local memory coupled to the data processor . . . and remotely located memory coupled to the data processor by a computer network.” Abstract.

*c. The Examiner has not provided a reasonable expectation of success in combining the teachings of Fletcher and Vishin*

The Examiner first brought forward this rejection in the November 2006 Office Action. Because this rejection is based on a pre-KSR obviousness rationale, a reasonable expectation of success is required. See, e.g., Examination Guidelines for Determining Obviousness Under 35 U.S.C. 103 in View of the Supreme Court Decision in *KSR Int’l Co. v. Teleflex Inc.*, 72 Fed. Reg. 57526, 57528, 57534 (Oct. 10, 2007).

As recited in lines 5-7 of Claim 1, the “*answer cache*” of Claim 1 can be examined by computer instructions to “*translate a domain name into DNS information*.” It is unclear to the Appellant how the RTL160 containing partial physical memory addresses taught by *Vishin* could be used to translate a domain name into DNS information. As

such, the proposed combination does not appear to have a reasonable expectation of success.

The Examiner responded to this argument by stating that “*Fletcher* does teach that computer instruction configured to translate a domain name into DNS information by examining the answer cache.” June 2007 Final Action, 9.

The Appellant maintains that while *Fletcher* teaches a computer instruction configured to translate a domain name into DNS information by examining the answer cache, it is unclear as to how the computer instructions of *Fletcher* can incorporate the RTL60 of *Vishin* with a reasonable expectation of success.

*d. The Examiner is improperly modifying the teachings of Vishin*

At least because there is no reasonable expectation of success, the Examiner is improperly modifying the teachings of *Vishin* in a manner unsupported by either reference in the rejection of claim 1. The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. *In re Mills*, 916 F.2d 680 (Fed. Cir. 1990).

Specifically, the Examiner appears to be suggesting that the teachings of *Vishin* be changed substantially in manners unsupported by either reference in order to fit the needs of the Examiner’s rejection, and that these changed teachings be added to *Fletcher*. Namely, the Examiner is suggesting that the RTL60 be modified in at least two ways. First, that the RTL60 be modified to translate a domain name into DNS information rather than “mapping between a range of physical addresses and a corresponding range of remote physical address.” *Vishin*, Abstract. Second, the Examiner is suggesting that the RTL60 is a flat data structure. As noted previously, the Appellant has reviewed the portions of *Vishin* previously cited by the Examiner, notably the abstract and FIG. 5, as well as the rest of the patent and is still unable to identify a hash table or other flat data structure in the cited reference. The Examiner did not respond to Appellant’s multiple requests for a specific teaching in *Vishin* of a “hash table” or “flat data structure.”

The Examiner has asserted that the motivation for modifying *Vishin* is found in the knowledge generally available to one skilled in the art. June 2007 Final Action, 9. “Therefore, in formulating a rejection under 35 U.S.C. §103(a) based upon a combination of prior art elements, it remains necessary to identify **the reason why** a person of ordinary skill in the art would have combined the prior art elements in the manner claimed.” *Memorandum regarding the Supreme Court decision on KSR Int’l Co., v. Teleflex, Inc.*, 2 (May 3, 2007) (Margaret A. Focarino) (emphasis added). As has been held by the Courts, “to establish a prima facie case, the USPTO may not rely on unsupported assertions about the level of ordinary skill in the art or bare conclusions that one of ordinary skill could apply such skill to obtain the claimed invention.” *In re Sun*, 31 USPQ2d 1451, 1456 (Fed. Cir. 1994)(Mayer, J. concurring).

The Appellant respectfully avers that a motivation “found in the knowledge generally available to one of ordinary skill in the art,” does not meet the threshold requirement to provide a prima facie case for a rejection under 35 U.S.C. §103(a) according to current practice. More specifically, the Examiner has not provided a **reason why** the knowledge generally available to one of ordinary skill in the art would motivate such a person to make the proposed modification to the teachings of *Vishin*.

*e. The Examiner has not provided a sufficient motivation to combine  
Fletcher and Vishin*

A prima facie case for rejection under §103(a) requires some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. (*In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). See MPEP § 2143 - § 2143.03.)

As a motivation to combine the cited art, the Examiner has asserted that “[b]y using the hash table (i.e. the flat data structure) in the answer cache as taught by *Vishin*, it reduces the number of memory accesses and as a result of that, it is faster than the lookup in the tree structure.” Notwithstanding that the Appellant is still unable to



identify a hash table or other flat data structure in *Vishin*, it is unclear how a buffer, even assuming *arguendo* a flat data structure, configured to translate virtual addresses into physical addresses would reduce the number of memory accesses used to translate a domain name into DNS information. Since *Vishin* does not teach the use of a “hash table,” any speed advantage that could theoretically be achieved therefrom cannot provide the basis for combining *Fletcher* and *Vishin*.

Further, the Appellant is unable to find any support within the cited art that use of the RTLB table of *Vishin* “reduces the number of memory accesses” as suggested by the Examiner. To the contrary, the use of a table for translating a physical memory address would appear to result in a greater number of memory accesses because the RTLB table must be accessed first to translate a virtual memory address into a physical memory address, rather than accessing the physical memory directly.

Rather, the Examiner appears to have defined a problem (lacking a hash table) in terms of its solution (adding a hash table). As stated in *In re Beattie*, 974 F.2d 1309, 1312 (Fed. Cir. 1992) “[d]efining the problem in terms of its solution reveals improper hindsight in the selection of the prior art relevant to obviousness.” The Appellant, therefore, requests that the Examiner provide either a motivation to combine the cited art from within the cited art or other evidence that the suggested motivation would be known to one of ordinary skill in the art at the time of the invention.

The Examiner has asserted that the motivation for combining the teachings of *Vishin* and *Fletcher* is found in the knowledge generally available to one skilled in the art. The Appellant again respectfully avers that a motivation “found in the knowledge generally available to one of ordinary skill in the art,” does not meet the threshold requirement to provide a prima facie case for a rejection under 35 U.S.C. §103(a) according to current practice. See, e.g., *Memorandum regarding the Supreme Court decision on KSR Int’l Co., v. Teleflex, Inc.*, 2 (May 3, 2007) (Margaret A. Focarino). More specifically, the Examiner has not provided a **reason why** the knowledge generally available to one of ordinary skill in the art would motivate such a person to make the proposed combination.

Claims 1, 11, 13, 17, 26, 34 and 38-39 are believed to be allowable for at least the above reasons.

**3. Present arguments with respect to claims 11, 13, 17, 38, and 39 with regard to the elements not included in claim 1**

The Examiner has rejected claims 11, 13, 17, 38, and 39 "based on the same rationale as the rejection of claim 1." November 2006 Office Action, 3; June 2007 Final Action, 3. To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981 (CCPA 1974).

Claim 11 includes additional elements not included in claim 1, such as:

- a code segment configured to examine a first cache to find the DNS information, the first cache including a flat data structure and configured to store the DNS information or a pointer to the DNS information; and
- a code segment configured to initiate a search of a second cache if the DNS information is not found by examining the first cache, the second cache configured to store data referring to further locations on a computer network wherein the DNS information may be found.

Claim 13 includes additional elements not included in claim 1, such as "means for examining a first cache to find the DNS information," and "means for searching a second cache ... the second cache configured to store data referring to further locations on the computer network wherein the DNS information may be found."

Claim 17 includes additional elements not included in claim 1, such as "a caching server including a first data structure configured for translating the domain name into DNS information, and means for examining the first data structure in a time that is essentially constant as a function of a number of labels comprising the domain name."

Claim 38 includes additional elements not included in claim 1, such as "wherein the answer cache is configured to store answer information and the referral cache is configured to store referral information."

Claim 39 includes additional elements not included in claim 1, such as “wherein the answer cache is configured to store answer information and the referral cache is configured to store referral information, and the answer cache and the referral cache have different data structures.”

Because these elements are not addressed in the rejection of claim 1, the Examiner has not provided a *prima facie* case of obviousness.

**B. Claims 2, 19, 25, 29, and 35**

**1. Summary of Prosecution with Respect to Claims 2, 19, 25, 29, and 35**

In the November 2006 Office Action, the Examiner rejected claims 2, 19, 25, 29, and 35. The Examiner based the rejection on an analysis of dependent claim 2. Page 3. Claims 25, 29, and 35 were “also rejected based on the same rationale as claim 2.”

November 2006 Office Action, 4. Claim 19 was rejected on the same rationale as the rejections of claims 1 and 2. Claim 2 recites, “[t]he caching server of claim 1, wherein the flat data structure is a hash table.”

In the April 2007 Response, the Appellant remarked that:

In rejecting Claim 2, the Examiner states “*Vishin* teaches that the flat data structure is a hash table (i.e. 122 in Fig. 5)(e.g. see the abstract and Fig. 5).” The Applicant traverses this statement. First, the Applicant is unable to identify any teaching in *Vishin* that the RTL160 includes a hash table. To the contrary, as illustrated in *Vishin* FIG. 6, the RTL160 is taught to include an index column (labeled “index”) having a sequential series of numbers (labeled “0 ... 31”). Thus, even if one were to assume for the sake of argument that the RTL160 of *Vishin* was a flat data structure, this flat data structure is clearly not a hash table.

Page 18. In the June 2008 Final Action, the Examiner merely repeated the rejection of claim 2 without addressing the above remarks. Page 3. In the August 2007 Response, the Appellant repeated the above remarks. In light of at least the above, the Appellant respectfully requested that the Examiner withdraw the finality of the June 2007 Final Action. Pages 21-22.

In the August 2007 Advisory Action, the Examiner stated:

Examiner would like to point out to Applicant that all arguments were responded/answered in the last office action mailed out on June 08, 2007 (See remarks section on page 8). Since some of the claims are rejected based on the same rationale as the rejection of other claim(s) (for example, claims 25, 29 and 35 are rejected based on the same rationale as rejection of claim 2), the response to the arguments is not repeated for each of these claims.

Page 3. The Appellant respectfully traverses because the Appellant is unable to identify *any* response to the above argument with respect to claim 2 in the June 2007 Final Action.

## **2. Present Arguments with Respect to Claims 2, 19, 25, 29, and 35**

The Examiner asserts that “*Vishin* teaches that the flat data structure is a hash table.” November 2006 Office Action, 3; June 2008 Final Action, 3. To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981 (CCPA 1974).

As noted throughout, the Appellant is unable to identify any teaching in *Vishin* that the RTL160 includes a hash table. To the contrary, as illustrated in *Vishin*, the RTL160 is taught to include an index column (labeled “index”) having a sequential series of numbers (labeled “0 ... 31”). FIG. 6. Thus, even if one were to assume for the sake of argument that the RTL160 of *Vishin* was a flat data structure, this flat data structure is clearly not a hash table.

Appellant believes claims 2, 19, 25, 29, and 35 are allowable at least because *Vishin* does not teach a hash table as recited in claim 2. Further Appellant believes claims 2, 25, 29, and 35 are allowable for at least the same reasons as the claims from which they depend. Appellant believes independent claim 19 is additionally allowable for the reasons discussed in connection with claim 1.

### **C. Claims 3, 5, 20-23, and 27-28**

#### **1. Summary of Prosecution with Respect to Claims 3, 5, 20-23, and 27-28**

In the November 2006 Office Action, the Examiner rejected claims 3, 5, 20-23, and 27-28. The Examiner based the rejection on an analysis of dependent claim 3. Page 4. Claims 5, 20-23, and 27-28 were “also rejected based on the same rationale as claim 3.” November 2006 Office Action, 4. Claim 3 recites, “[t]he caching server of claim 1, wherein the flat data structure includes pointers to a tree data structure.”

The Examiner rejected claim 3 based on the rationale that:

*Fletcher* teaches that when the requested address information is not found at the terminal, the query from the local host is forwarded to the communication network (i.e. see paragraph [0008]). Therefore, the pointer/link has to be inherently stored/present in the local terminal cache that points to the remote host for the requested information.

November 2006 Office Action, 4.

The Appellant responded to the rejection asserting that the Examiner was improperly relying on a theory of inherency with respect to the teachings of *Vishin*.

April 2007 Response, 19. The Appellant stated:

the system of *Vishin* is a hardware based system in which communication can take place purely based on hardwired connections such as the "Match Signals" and "Filter Selector 164" illustrated in FIG. 6 of *Vishin*. It is, therefore, not inherent that the RTL160 of *Vishin*, which the Examiner suggests teaches an answer cache, include pointers as recited in Claim 13.

Further, even if for the sake of argument, it were assumed that the RTL160 of *Vishin* included a pointer, there does not appear to be any teaching of "DNS information" in *Vishin*, much less that these hypothetical pointers point to DNS information. To the contrary, as shown in FIG. 6 of *Vishin*, the entries in the RTL160 are coupled to RPPA entries in SRAM 166. *Vishin* does not appear to include any suggestion that these RPPA entries include DNS information. The Applicant, therefore, requests that the Examiner specifically point out a teaching within *Vishin* that RTL160 includes "DNS information or a pointer to the DNS information" as recited in Claim 3, or allow Claim 3.

In rejecting Claim 3, the Examiner further states "*Fletcher* teaches the further limitation of pointers pointing to a tree data structure (e.g. see paragraph [0005])." However, the Applicant respectfully points out that Claim 3 recites that the "flat data structure includes pointers to a tree data structure" and the Examiner is citing *Vishin*, not *Fletcher*, as teaching the "flat data structure." Specifically, the Examiner cites the RTL160 of *Vishin* as teaching the flat data structure of Claim 13, and the RTL160 is not taught to include "pointers to a tree data structure."

April 2007 Response, 19-20. In the June 2008 Final Action, the Examiner merely repeated the rejection of claim 3 without addressing the above remarks. Page 4. In the August 2007 Response, the Appellant repeated the above remarks. In light of at least the above, the Appellant respectfully requested that the Examiner withdraw the finality of the June 2007 Final Action. Page 24.

In the August 2007 Advisory Action, the Examiner stated:

Examiner would like to point out to Applicant that all arguments were responded/answered in the last office action mailed out on June 08, 2007 (See remarks section on page 8).

Page 3. The Appellant respectfully traverses because the Appellant is unable to identify any response to the above argument with respect to claim 3 in the June 2007 Final Action.

## 2. Present Arguments with Respect to Claims 3, 5, 20-23, and 27-28

The Examiner states that:

*Fletcher* teaches that when the requested address information is not found at the terminal, the query from the local host is forwarded to the communication network (i.e. see paragraph [0008]). Therefore, the pointer/link has to be inherently stored/present in the local terminal cache that points to the remote host for the requested information.

November 2006 Office Action, 4; June 2007 Final Action, 4.

“In relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art” citing *Ex parte Levy*, 17 USPQ2d 1461, 1464 (Bd. Pat. App. & Inter. 1990).

The system of *Vishin* is a hardware based system in which communication can take place purely based on hardwired connections such as the “Match Signals” and “Filter Selector 164” illustrated in FIG. 6 of *Vishin*. It is, therefore, not inherent that the RTL60 of *Vishin*, which the Examiner suggests teaches an answer cache, include pointers as recited in claim 3.

Further, even if for the sake of argument it were assumed that the RTL60 of *Vishin* included a pointer, there does not appear to be any teaching of “DNS information” in *Vishin*, much less that these hypothetical pointers point to DNS information. To the contrary, as shown in FIG. 6 of *Vishin*, the entries in the RTL60 are coupled to RPPA entries in SRAM 166. *Vishin* does not appear to include any suggestion that these RPPA entries include DNS information.

In rejecting claim 3, the Examiner further states “*Fletcher* teaches the further limitation of pointers pointing to a tree data structure (e.g. see paragraph [0005]).” However, the Appellant respectfully points out that claim 3 recites that the “flat data structure includes pointers to a tree data structure” and the Examiner is citing *Vishin*, not *Fletcher*, as teaching the “flat data structure.” Specifically, the Examiner cites the RTL60 of *Vishin* as teaching the flat data structure of claim 13, and the RTL60 is not taught to include “pointers to a tree data structure.”

For at least these reasons the Appellant believes claims 3, 5, 20-23, and 27-28 are allowable over *Fletcher* in view of *Vishin*.

**3. Present arguments with respect to claims 3, 5, 20-23, and 27-28 with regard to the elements not included in claim 1**

The Examiner has rejected claims 5, 20-23, and 27-28 “based on the same rationale as the rejection of claim 1.” November 2006 Office Action, 3; June 2007 Final Action, 3. To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981 (CCPA 1974).

Claim 5 includes additional elements not included in claim 3, such as “the tree data structure is included in the referral cache.”

Claim 20 includes additional elements not included in claim 3, such as “wherein the hash table is configured to store the pointer to the answer information.”

Claim 23 includes additional elements not included in claim 3, such as “wherein the tree data structure is configured to store pointers to referral data.”

Claim 27 includes additional elements not included in claim 3, such as “wherein the flat data structure is configured to store the answer information.”

Claim 28 includes additional elements not included in claim 3, such as “wherein the flat data structure is configured to store a pointer to the answer information.”

Because these elements are not addressed in the rejection of claim 3, the Examiner has not provided a case of *prima facie* obviousness. Claims 21 and 22 are allowable for at least the same reasons as Claim 19, from which they depend.



**D. Claim 4**

**1. Summary of Prosecution with Respect to Claim 4**

In the November 2006 Office Action, the Examiner rejected claim 4. Page 4. Claim 4 recites, “[t]he caching server of claim 1, wherein the flat data structure includes pointers to a tree data structure, and the tree data structure is configured to store answer information and referral information.”

The Examiner rejected claim 4 based on the rationale that “*Fletcher* teaches that the tree data structure (i.e., the hierarchical structure) is configured to store answer information and referral information (e.g. see paragraphs [0005]-[0006]).”

November 2006 Office Action, 4.

The Appellant responded to the rejection asserting that:

In the combination suggested by the Examiner, it appears that the flat data structure is suggested as being taught by the RTL160 of *Vishin*, however, the RTL160 of *Vishin* is not taught to include pointers to a tree data structure, much less a tree data structure “*configured to store answer information and referral information*,” as recited in Claim 4. As discussed above the RTL160 of *Vishin* is taught to include information for translating remote physical memory addresses. The Applicant, therefore, requests that the Examiner specifically point out a teaching that the RTL160 of *Vishin* includes pointers to a tree data structure and that that tree data structure includes “*answer information and referral information*,” or allow Claim 4.”

April 2007 Response, 22. In the June 2008 Final Action, the Examiner merely repeated the rejection of claim 4 without replying to the above remarks. Page 4. In the August 2007 Response, the Appellant repeated the above remarks. In light of at least the above, the Appellant respectfully requested that the Examiner withdraw the finality of the June 2007 Final Action. Pages 25-26.

In the August 2007 Advisory Action, the Examiner stated:

Examiner would like to point out to Applicant that all arguments were responded/answered in the last office action mailed out on June 08, 2007 (See remarks section on page 8).

Page 3. The Appellant respectfully traverses because the Appellant is unable to identify *any* response to the above argument with respect to claim 4 in the June 2007 Final Action.

## 2. Present Arguments with Respect to Claim 4

The Examiner states that “*Fletcher* teaches that the tree data structure (i.e., the hierarchical structure) is configured to store answer information and referral information (e.g. see paragraphs [0005]-[0006]).” November 2006 Office Action, 4; June 2007 Final Action, 4.

However, claim 4 more fully recites, “wherein the flat data structure includes pointers to a tree structure, and the tree data structure is configured to store answer information and referral information.” Because claim 4 includes the element of claim 3, “wherein the flat data structure includes pointers to a tree data structure,” claim 4 is allowable for at least the same reasons as claim 3.

In the combination suggested by the Examiner, it appears that the flat data structure is suggested as being taught by the RTL160 of *Vishin*, however, the RTL160 of *Vishin* is not taught to include pointers to a tree data structure, much less a tree data structure “configured to store answer information and referral information,” as recited in Claim 4. As discussed above the RTL160 of *Vishin* is taught to include information for translating virtual memory addresses into remote physical memory addresses rather than translating domain names into DNS information.

For at least these reasons the Appellant believes claim 4 is allowable over *Fletcher* in view of *Vishin*.

### E. Claims 6-9, 12, 16, 18, 24, 31-32, and 41-43

As a dependent claim incorporates by reference all the limitations of the claim from which it depends (see 35 U.S.C. § 112, ¶ 4), the rejection of claims 6-9, 12, 16, 18, 24, 31-32, and 41-43 should be overturned for at least the same reasons as the claims from which they depend.

**F. Claims 14-15 and 30**

**1. Summary of Prosecution with Respect to Claims 14-15 and 30**

Claim 14 recites, "The computer network of claim 13, further including means for storing data in the first cache such that a time required to examine the first cache is essentially constant as a function of a number of labels comprising the domain name."

Claim 15 recites "The computer network of claim 13, further including means for storing data in the first cache such that a time required to examine the first cache is essentially constant as a function of a size of the first cache."

In the November 2006 Office Action, the Examiner rejected claims 14 and 15.

Page 5-6. In rejecting claims 14 and 15, the Examiner states:

*Fletcher* teaches means for storing data in the first cache such that a time required to examine the first cache is essentially constant as a function of a number of labels comprising the domain name, i.e. the first cache is the local cache, which uses the flat data structure and since the number of cache entries to search in this flat data structure local cache is fixed/constant as a function of (i) a number of labels comprising the domain name and (ii) a size of the first/local cache (e.g. see paragraph [0008]).

November 2006 Office Action, 5-6. Claim 30 was rejected on the same rationale as the rejection of claims 14-15.

The Appellant responded to the rejection asserting that:

**First**, in the rejection of Claim 1 the Examiner admits that *Fletcher* does not teach an answer cache configured to store data in a flat data structure. This admission appears to be directly contradictory to the above statement made in rejecting Claims 14 and 15. The Applicant agrees that *Fletcher* does not teach a flat data structure. The Applicant, therefore, requests that the Examiner withdraw the rejections of Claims 14 and 15.

**Second**, even assuming for the sake of argument that *Fletcher* taught a flat data structure, there is no indication that such a data structure would have a structure that satisfies the conditions that "a time required to examine the first cache is essentially constant as a function of a number of labels comprising the domain name" or "a time required to examine the first cache is essentially constant as a function of a size of the first cache," as recited in Claims 14 and 15, respectively. The Applicant respectively points out that these are not normal characteristics of flat data structures. To the contrary, one of ordinary skill in the art would expect that a time required to search a flat data would grow linearly with the size of the cache. The Applicant, therefore, requests that the Examiner

specifically point out how *Fletcher* teaches these limitations of Claims 14 and 15, or allow Claims 14 and 15.

April 2007 Response, 23-24. In the June 2008 Final Action, the Examiner merely repeated the rejection of claims 14 and 15 without replying to the above remarks. Page 5-6. In the August 2007 Response, the Appellant repeated the above remarks. In light of at least the above, the Appellant respectfully requested that the Examiner withdraw the finality of the June 2007 Final Action. Page 28.

In the August 2007 Advisory Action, the Examiner stated:

Examiner would like to point out to Applicant that all arguments were responded/answered in the last office action mailed out on June 08, 2007 (See remarks section on page 8).

Page 3. The Appellant respectfully traverses because the Appellant is unable to identify any response to the above argument with respect to claims 14-15 in the June 2007 Final Action.

## **2. Present Arguments with Respect to Claims 14-15 and 30**

The Examiner states that

*Fletcher* teaches means for storing data in the first cache such that a time required to examine the first cache is essentially constant as a function of a number of labels comprising the domain name, i.e. the first cache is the local cache, which uses the flat data structure and since the number of cache entries to search in this flat data structure local cache is fixed/constant as a function of (i) a number of labels comprising the domain name and (ii) a size of the first/local cache (e.g. see paragraph [0008])

November 2006 Office Action, 5-6; June 2007 Final Action, 5-6.

In the rejection of Claim 1, the Examiner admits that *Fletcher* does not teach an answer cache configured to store data in a flat data structure. This admission appears to be directly contradictory to the above statement made in rejecting Claims 14 and 15. The Appellant agrees that *Fletcher* does not teach a flat data structure.

Even assuming for the sake of argument that *Fletcher* taught a flat data structure, there is no indication that such a data structure would have a structure that satisfies the conditions that "a time required to examine the first cache is essentially constant as a function of a number of labels comprising the domain name" or "a time required to

examine the first cache is essentially constant as a function of a size of the first cache,” as recited in claims 14 and 15, respectively. These are not normal characteristics of flat data structures. To the contrary, one of ordinary skill in the art would expect that a time required to search a flat data would grow linearly with the size of the cache.

Therefore, Appellant believes claims 14-15 and 30 are allowable for at least the above reasons.

**G. Claims 33, 36-37, and 40**

**1. Summary of Prosecution with Respect to Claims 33, 36-37, and 40**

Claim 33 recites:

A method of storing data in a cache, the method comprising:  
requesting DNS information;  
receiving data in response to the request;  
classifying the response received; and  
storing the data received in either a referral cache or an answer cache  
based on the classification.

Regarding claim 33, in the November 2006 Office Action, the Examiner states:

*Fletcher* teaches ... requesting DNS information; receiving data in response to the request; classifying the response received; and storing the data received in either a referral cache or an answer cache based on the classification (e.g. see paragraphs [0005] and [0008]).

November 2006 Office Action, 6. Claim 36 was rejected on the same rationale as the rejection of claim 33.

The Appellant responded to the rejection asserting that:

The Applicant has reviewed *Fletcher*, and in particular those paragraphs cited by the Examiner, however, the Applicant is unable to identify any teaching of “classifying the response received; and storing the data received in either a referral cache or an answer cache based on the classification,” as recited in Claim 33.

April 2007 Response, 24. In the June 2008 Final Action, the Examiner merely repeated the rejection of claims 33 and 36 without replying to the above remarks. Page 6. In the August 2007 Response, the Appellant repeated the above remarks. In light of at least the above, the Appellant respectfully requested that the Examiner withdraw the finality of the June 2007 Final Action. Page 29.

In the August 2007 Advisory Action, the Examiner stated:

Examiner would like to point out to Applicant that all arguments were responded/answered in the last office action mailed out on June 08, 2007 (See remarks section on page 8).

Page 3. The Appellant respectfully traverses because the Appellant is unable to identify *any* response to the above argument with respect to claims 33 and 36 in the June 2007 Final Action.

## **2. Present Arguments with Respect to Claims 33 and 36**

The Examiner states that "*Fletcher* teaches ... requesting DNS information; receiving data in response to the request; classifying the response received; and storing the data received in either a referral cache or an answer cache based on the classification (e.g. see paragraphs [0005] and [0008])."

The Appellant has reviewed *Fletcher*, and in particular those paragraphs cited by the Examiner, however, the Appellant is unable to identify any teaching of "classifying the response received; and storing the data received in either a referral cache or an answer cache based on the classification," as recited in Claim 33.

Thus, the Appellant believes claims 33 and 36 are patentable. Further, claim 37 is allowable for at least the same reasons as claim 33 from which it depends. Further, Appellant believe claim 40 is patentable for at least the reasons discussed in connection to the rejections of claim 1 and claim 33.

## **II. REJECTION OF CLAIM 10 (*FLETCHER* IN VIEW OF *VISHIN*, FURTHER IN VIEW OF *RAMANATHAN*)**

As a dependent claim incorporates by reference all the limitations of the claim from which it depends (see 35 U.S.C. § 112, ¶ 4), the rejection of claims 10 should be overturned for at least the same reasons as claim 1.

Respectfully submitted,  
Andreas Gustafson

November 9, 2007

By: Karen Kaufman  
Karen Kaufman (Reg. No. 57,239)  
**CARR & FERRELL LLP**  
2200 GENG ROAD  
PALO ALTO, CA 94303  
T: 650.812.3400  
F: 650.812.3444

ATTORNEY FOR THE APPELLANT  
AND REAL-PARTY-IN-INTEREST

**CLAIMS APPENDIX**  
**(37 C.F.R. § 41.37(C)(1)(viii))**

The claims involved in the present appeal are as follows:

1. A caching server comprising:
  - an answer cache configured to access answer information through a flat data structure;
  - a referral cache configured to store referral information; and
  - computer instructions configured to translate a domain name into DNS information by examining the answer cache and, responsive to the results of examining the answer cache, examining the referral cache.
2. The caching server of claim 1, wherein the flat data structure is a hash table.
3. The caching server of claim 1, wherein the flat data structure includes pointers to a tree data structure.
4. The caching server of claim 1, wherein the flat data structure includes pointers to a tree data structure, and the tree data structure is configured to store answer information and referral information.
5. The caching server of claim 1, wherein the flat data structure includes pointers to a tree data structure, and the tree data structure is included in the referral cache.



6. The caching server of claim 1, wherein the caching server is also an authoritative server.
7. The caching server of claim 1, wherein the caching server is also a web server.
8. The caching server of claim 1, wherein the referral cache is further configured to store the referral information in a hierarchical data structure.
9. The caching server of claim 1, wherein the DNS information includes an IP address.
10. The caching server of claim 1, wherein the DNS information includes an MX record.
11. A computer readable medium having stored thereupon computer code configured to determine DNS information associated with a domain name, the computer code comprising:
  - a code segment configured to receive a request for the DNS information corresponding to a domain name;
  - a code segment configured to examine a first cache to find the DNS information, the first cache including a flat data structure and configured to store the DNS information or a pointer to the DNS information; and
  - a code segment configured to initiate a search of a second cache if the DNS information is not found by examining the first cache, the second cache

configured to store data referring to further locations on a computer network wherein the DNS information may be found.

12. The computer readable medium of claim 11, wherein the DNS information includes an IP address.

13. A computer network comprising:

means for receiving a request for DNS information corresponding to a domain name;

means for examining a first cache to find the DNS information, the first cache configured to store the DNS information or a pointer to the DNS information; and

means for searching a second cache if the DNS information is not found by examining the first cache, the second cache configured to store data referring to further locations on the computer network wherein the DNS information may be found.

14. The computer network of claim 13, further including means for storing data in the first cache such that a time required to examine the first cache is essentially constant as a function of a number of labels comprising the domain name.

15. The computer network of claim 13, further including means for storing data in the first cache such that a time required to examine the first cache is essentially constant as a function of a size of the first cache.
16. The computer network of claim 14, wherein the DNS information includes an IP address.
17. A computer network comprising:
- a computing system configured to access a component of the computer network using a domain name;
  - a caching server including a first data structure configured for translating the domain name into DNS information, and means for examining the first data structure in a time that is essentially constant as a function of a number of labels comprising the domain name; and
  - a second data structure configured for translating the domain name into DNS information.
18. The computer network of claim 17, wherein the DNS information includes an IP address or an MX record.
19. A method of determining DNS information, the method comprising:
- receiving a request for DNS information corresponding to a domain name;

examining an answer cache for answer information, the answer cache including a hash table configured to store the answer information or to store a pointer to the answer information; and searching a tree data structure if the DNS information is not found by examining the answer cache.

20. The method of claim 19, wherein the hash table is configured to store the pointer to the answer information.

21. The method of claim 19, wherein the answer cache does not include a tree data structure.

22. The method of claim 19, wherein the tree data structure is configured to store referral data and is included in a referral cache.

23. The method of claim 19, wherein the tree data structure is configured to store pointers to referral data.

24. The method of claim 19, wherein the DNS information includes an IP address.

25. The method of claim 19, wherein the hash table is configured to store the answer information.

26. A method of determining DNS information, the method comprising:  
receiving a request for DNS information corresponding to a domain name;

examining an answer cache to find answer information, responsive to the  
received request, the answer cache including a flat data structure; and  
responsive to the examination of the answer cache, searching a referral cache.

27. The method of claim 26 wherein the flat data structure is configured to store the answer information.
28. The method of claim 26, wherein the flat data structure is configured to store a pointer to the answer information.
29. The method of claim 26, wherein the flat data structure is a hash table.
30. The method of claim 26, wherein a time required to examine the answer cache is essentially constant as a function of a number of labels comprising the domain name and essentially constant as a function of a size of the answer cache.
31. The method of claim 26, wherein the referral cache includes a hierarchical data structure.
32. The method of claim 26, wherein the DNS information includes an IP address.
33. A method of storing data in a cache, the method comprising:
  - requesting DNS information;
  - receiving data in response to the request;
  - classifying the response received; and

storing the data received in either a referral cache or an answer cache based on the classification.

34. The method of claim 33, wherein the answer cache includes a flat data structure.

35. The method of claim 33, wherein the answer cache includes a hash table.

36. The method of claim 33, wherein the response received is stored in a caching server.

37. The method of claim 33, wherein the DNS information includes a numerical address.

38. The method of claim 33, wherein the answer cache is configured to store answer information and the referral cache is configured to store referral information.

39. The method of claim 33, wherein the answer cache is configured to store answer information and the referral cache is configured to store referral information, and the answer cache and the referral cache have different data structures.

40. A method of caching DNS information, the method comprising:  
requesting DNS information;  
receiving data in response to requesting DNS information;  
classifying the response received as an answer response or a referral response;  
storing the response received in either a referral cache or an answer cache based on the classification, the answer cache including a flat data structure;  
receiving a request for DNS information corresponding to a domain name;

examining the answer cache to find answer information, responsive to the  
received request; and  
responsive to the examination of the answer cache, searching the referral cache.

41. The method of claim 40, wherein the referral cache includes a hierarchical data structure.
42. The method of claim 40, wherein the received request for DNS information includes a request for an IP address.
43. The caching server of claim 1, wherein the referral cache is separate from the answer cache.

**EVIDENCE APPENDIX**  
**37 C.F.R. § 41.37(C)(1)(ix)**

No evidence submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132 has been presented or entered during prosecution of the present application. As such, no evidence under the aforementioned sections is presented or referenced herewith.



**RELATED PROCEEDINGS APPENDIX**  
**37 C.F.R. § 41.37(C)(1)(x)**

No related proceedings including appeals or interferences—either concluded, ongoing, or otherwise prospective—are known to the Appellant, real-party-in-interest, nor their agents and representatives. As such, no decisions or documentation related to such a proceedings is presented or referenced herewith.